CAPE: Encoding Relative Positions with Continuous Augmented Positional Embeddings

Anonymous Author(s)
Affiliation
Address
email

Abstract

Without positional information, attention-based transformer neural networks are permutation-invariant. Absolute or relative positional embeddings are the most popular ways to feed transformer models positional information. Absolute positional embeddings are simple to implement, but suffer from generalization issues when evaluating on sequences of different length than those seen at training time. Relative positions are more robust to length change, but are more complex to implement and yield inferior model throughput. In this paper, we propose an augmentation-based approach (CAPE) for absolute positional embeddings, which keeps the advantages of both absolute (simplicity and speed) and relative position embeddings (better generalization). In addition, our empirical evaluation on state-of-the-art models in machine translation, image and speech recognition demonstrates that CAPE leads to better generalization performance as well as increased stability with respect to training hyper-parameters.

1 Introduction

Transformers have been shown to be highly effective on problems involving sequential modeling, such as in machine translation (MT) [39] and natural language processing (NLP) [12, 34, 5]. Following its success on these tasks, the Transformer architecture raised immediate interest on other domains: automatic speech recognition (ASR) [13, 20], music generation [23], object detection [6], and finally image recognition [14, 38] and video understanding [4].

Two major components of the Transformer are the attention mechanism [2, 39] and the positional encoding [39, 36, 23, 10]. Without the latter, vanilla attention Transformers are invariant with respect to input token permutations (making "cat eats fish" and "fish eats cat" identical to the model). In the original Transformer publication, sinusoidal positional encoding were introduced [39]. Token positions were encoded in an absolute manner, which was sufficient to achieve state-of-the-art performance in numerous tasks. Performance issues were however later observed when dealing with long sequences not seen at training time [23, 10, 49, 28]. In fact, for most applications, relative positions between tokens are more relevant than absolute ones. A number of approaches were thus investigated to encode relative positions in an explicit form [36, 10, 23], leading to an improvement in modeling long sequences. However, all these approaches focus on modifying the attention mechanism and suffer from additional computational and memory cost [47]. Relative positional encoding is also notably hard to implement efficiently for multidimensional case, and recent advances in Transformer models for computer vision [6, 14, 38] still rely on learnable absolute positional encoding.

Instead of changing the Transformer attention mechanism, we propose to improve absolute sinusoidal positional encodings in two ways: a) instead of discrete positions, rely on continuous ones, which better match the continuous nature of image, sound or video data; b) preserve some information about relative token positions, via a specific noise augmentation approach for positional embeddings during

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

training. Dubbed continuous augmented positional embedding (CAPE), we empirically evaluate our approach to recent state-of-the art models in several application domains. We study generalization properties and introduce unique features unlocked by CAPE. The main contributions of this work are:

- new augmented continuous positional embedding (CAPE), which encodes some relative position information in a computationally efficient way, and improves generalization performance compared to other positional embeddings across a variety of domains: machine translation, image and speech recognition;
- a single vision Transformer (UniViT) trained with CAPE on the mix of different resolutions: it outperforms each single-resolution baseline, generalizes better to unseen resolutions and can naturally process images of any size;
- new CAPE-based adaptive training scheme for ASR that eliminates need for padding.

2 Related Works

Since the introduction of Transformers, many works have investigated ways to encode positional information. A detailed analysis of various positional embeddings is available for BERT architecture [40], where authors empirically relate properties of positional embeddings to performance on downstream NLP tasks. A recent study [24] (also focused on BERT) highlights the negative impact of spurious correlations between word and position embedding, and proposes to explicitly disentangle the contribution of positional and content embeddings in the attention mechanism. In contrast, our approach implicitly enforces this disentanglement by leveraging noise augmentation.

Systematic studies of positional embeddings in audio domain are scarce. Several ways to encode relative positions for Transformer-based speech recognition are compared in [41]. Experiments show that sinusoidal positional embeddings work no better than stacking consecutive frames at each time position (a particular form of convolution). We provide a more thorough evaluation of positional embeddings in ASR, over multiple datasets. We also show that embeddings obtained from a one-layer convolutional frontend benefits from adding positional information.

Transformers for computer vision applications are still in their early days, and most works rely on learnable absolute positional embeddings only [14, 38, 4, 1]. Several recent works complement the Transformer architecture with convolutional layers, either to improve the representation captured by tokens, or to induce a spacial relationship between tokens [18, 8, 43]. As we discuss later, this restricts flexibility of pure attention-based models. The work [18] suggests injecting learnable attention biases as an alternative mechanism to positional encoding. Evaluation of several positional encodings and their corresponding generalization has been done in a study [8] which is in line with our work. Convolutional elements were introduced in the Transformer architecture, leading to better generalization properties. In contrast, our experiments demonstrate that generalization can be achieved without architecture modification or convolutional inductive bias. Concerning video understanding, an evaluation of the impact of positional encoding was performed in [29]: according to the results, positional encoding-free architectures performed best. Other work [4] reports that adding absolute positional embeddings improves models performance, but contribution of encoding space and time vary between datasets.

As a summary, many positional embeddings variants were previously introduced, often modality-specific. In our cross-modal study we focus on generalization properties of different embeddings, and improve on sinusoidal positional embeddings, leading to a flexible Transformer architecture, with great generalization properties across a number of different tasks.

3 Theoretical Analysis of Sinusoidal Positional Embeddings

Originally positional K-dimensional embeddings were introduced in [39] as

$$E_{2k}(n) = \cos \omega_k n \qquad E_{2k+1}(n) = \sin \omega_k n \qquad \omega_k = 10000^{-2k/K} \qquad n \in \mathbb{Z}^+ \tag{1}$$

with $k = 1, 2, \ldots, K/2$ enumerating components for a token at position n. For simplicity of analysis, we consider a complex-valued representation of embeddings with half the number of components:

$$\{\mathbf{E}(n)\}_k = E_k(n) = e^{i\omega_k n}$$

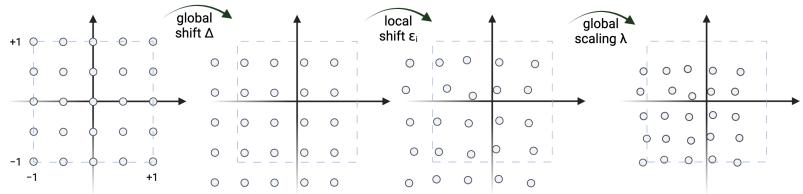

Figure 1: Example of CAPE's transformations for an image: patches positions are scaled to $[-1, 1] \times [-1, 1]$; a random global shift, local shift, and global scaling are then applied to the grid of positions.

This definition can be rewritten in a recursive manner, by introducing a unitary operator S:

$$\mathbf{E}(n+1) = S\,\mathbf{E}(n)\,, \qquad \text{with} \qquad \{S\,\mathbf{X}\}_k = X_k e^{i\omega_k} \tag{2}$$

Therefore, the embedding at position n contains sufficient information to compute the embedding of the next or previous positions, as applying S^m ($m \in \mathbb{Z}$) performs a relative shift: $\mathbf{E}(n+m) = S^m\,\mathbf{E}(n)$. Variation in ω_i ensures that no positions in the order $< 10^4$ are assigned similar embeddings. Before introducing noise augmentation techniques over positional embeddings, we revisit positional embedding parametrizations for different modalities.

Positional encoding for text For natural language processing it is common to split text into words, letters, syllables and other sub-units. Original sinusoidal positional embeddings enumerate these sub-units by their ordinal number n, a common choice that we follow.

Positional encoding for images In a framework where patch and image sizes may vary, we find that enumerating patches is not appropriate, as positional embeddings may greatly differ for different scales of different image, leading to generalization issues. In that perspective, we consider scaled coordinates x and y that span interval $[-1, +1]$. While previous works [14, 38] relied on absolute learnable positional embedding, we introduce the following K-component absolute sinusoidal positional embedding defined for each position $(x, y) \in \mathbb{R}^2$:

$$E_{2k}(x, y) = \cos \pi(w_{k,x}x + w_{k,y}y) \qquad E_{2k+1}(x, y) = \sin \pi(w_{k,x}x + w_{k,y}y)$$

$$w_{k,x} = 10^{2k/K} \cos k \qquad w_{k,y} = 10^{2k/K} \sin k$$

Following (2), this corresponds to introducing two commuting unitary operators S_x and S_y, for each unit shift in either direction of the plane. Choice of w is kept simple and deterministic while giving no preference to any direction on a plane.

Positional encoding for sound We propose to tie positional embeddings to timestamps in seconds. The embedding for a frame centered at t seconds is given by:

$$E_{2k}(t) = \sin \omega_k t \qquad E_{2k+1}(t) = \cos \omega_k t \qquad \omega_k = 30 \times 10000^{-2k/K}$$

The choice of ω_k corresponds to the scaled version of (1) and ensures that queries with 30ms specificity are possible even with minutes-long audio fragments.

4 Continuous Augmented Positional Embeddings (CAPE)

Regular sinusoidal positional embeddings lack regularization, leading to "in-domain" generalization issues as the model may start learning spurious correlations. For applications where inference input sizes may be significantly different than the training ones, "out-of-domain" issues may also arise, as positions rarely observed at training may lead to improper inference predictions. To ensure that the model does not learn spurious correlations of the content and position, we introduce the following noise augmentations for positional embeddings at training time.

Global shift Transform every embedding in a sequence using S^Δ operator with a global random shift from uniform zero-mean distribution $\Delta \sim \mathcal{U}(-\Delta_{max}, \Delta_{max})$:

$$\mathbf{E}'(n) = S^\Delta \mathbf{E}(n) \qquad \{S^\Delta\,\mathbf{X}\}_k = X_k e^{i\omega_k \Delta} \qquad \Delta \in \mathbb{R}$$

This modification smooths the absolute positional information, but relations (2) between embeddings still hold. This transformation can be rewritten as augmenting positions by a random shift before encoding with $\sin$ and $\cos$:

$$n_i' \leftarrow n_i + \Delta \qquad x_i' \leftarrow x_i + \Delta_x, \;\; y_i' \leftarrow y_i + \Delta_y \qquad t_i' \leftarrow t_i + \Delta \tag{3}$$

Local shift To further increase augmentation and prevent capturing spontaneous correlations, we additionally introduce local shifts from uniform zero-mean distribution $\epsilon_i \sim \mathcal{U}(-\epsilon_{max}, \epsilon_{max})$

$$n_i' \leftarrow n_i + \epsilon_i \qquad x_i' \leftarrow x_i + \epsilon_{x,i}, \;\; y_i' \leftarrow y_i + \epsilon_{y,i} \qquad t_i' \leftarrow t_i + \epsilon_i \tag{4}$$

Global scaling To prevent distances memorization, we also introduce random global scalings λ from $\log \lambda \sim \mathcal{U}(-\log \lambda_{max}, \log \lambda_{max})$

$$n_i' \leftarrow \lambda n_i \qquad x_i' \leftarrow \lambda x_i, \quad y_i' \leftarrow \lambda y_i \qquad t_i' \leftarrow \lambda t_i \tag{5}$$

At training time, computing our continuous augmented positional embedding (CAPE) is performed through four steps: i) mean-normalization of positions, ii) global shift (3), iii) local shift (4), and iv) global scaling (5). At inference time, only i) mean-normalization of positions is performed.

5 Experiments

5.1 Image Recognition

We evaluate CAPE embedding empirically on the recently proposed vision Transformer (ViT) [14, 38] for image recognition. These approaches rely on learnable absolute positional embedding (*abspos*) for both class token and patches, and train ViT models on 224^2 images[1] with 16^2 patches. To further improve model quality, [38] performs fine-tuning on images with higher resolution 384^2. The grid of positional embeddings is then upsampled.

Data and ViT Models All experiments were performed on the ImageNet [11, 35] dataset. We report top-1 and top-5 accuracy on both the ImageNet validation set and ImageNet-v2-{a,b,c} [33] test sets. The same convolution-free architecture, identical to the one proposed by [14] (ViT-B) and used by [38] (referred as DeiT-B) is chosen for all experiments. A ViT-B/DeiT-B baseline is trained with *abspos* on 224^2 images, carefully following Section 6 from [38].[2] The *exact same training configuration* was used for evaluating other positional embeddings: *only* the positional embedding layer was changed. We evaluated both the proposed sinusoidal positional embedding (*sinpos*), and CAPE ($\Delta_{max} = 0.5$, $\epsilon_{max} = 1/N$ and $\lambda_{max} = 1.4$). As a control experiment we also train a model without any positional embedding (*nopos*), which can be interpreted as a 'bag of ~~words~~ patches', as no patch position information is available. We also train models with different positional embeddings on either 160^2 or 384^2 images. The whole training configuration remains the same as for training on 224^2 images, except for the positional embedding layer. For all models trained on 224^2 images we additionally perform fine-tuning on images with higher resolution 384^2, following [38]. Finally, both *sinpos* and CAPE use learnable absolute positional embedding for the class token.

Evaluation To study generalization performance when image sizes vary, we evaluate all models on different resolutions: 160^2, 224^2, 384^2 and 672^2. When evaluating on resolutions different from the training one, bicubic interpolation is applied to *abspos* embeddings[3] as was justified in [38]. In contrast, *sinpos* and CAPE approaches can ingest any image resolution, thanks to the continuous nature of their positional embeddings.

5.1.1 Results

In Figure 2 we compare generalization performance of models trained with different positional embeddings on 224^2 images (solid). Both proposed *sinpos* and CAPE approaches perform at least

[1] In the following, we denote the resolution $N \times N$ as N^2.

[2] Initialization is set to the truncated normal distribution, Rand-Augment [9], Mixup [46] and Cutmix [44], random erasing [48] and repeated augmentation [22, 3] are used as data augmentations; models are trained with AdamW optimizer for 300 epochs.

[3] Not to the class token embedding.

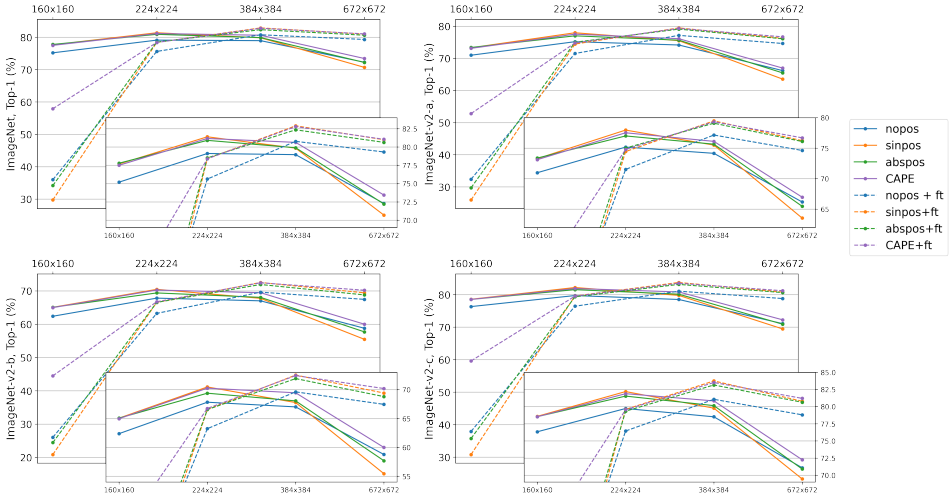

Figure 2: Top-1 accuracy on ImageNet and ImageNet-v2 for ViT models trained with different positional embeddings on 224^2 resolution (solid) and further fine-tuned on 384^2 (dashed, '+ft'). The full list of top-1 and top-5 accuracy can be found in Appendix B.1, Tables 3 and 4.

as well, if not better, than the *abspos* approach on the same-as-train resolution. When performing inference on resolutions different than the training one, CAPE performs best, notably outperforming *sinpos* on high (672^2) and low (160^2) resolutions. On 160^2 and 384^2 resolutions CAPE trained on 224^2 resolution performs similarly than *abspos* trained on corresponding 160^2 or 384^2 inference resolutions (the latter results being reported in Figure 3). This confirms good generalization properties of CAPE on image resolutions unseen at training time.

Abspos fine-tuned on a higher resolution (384^2) improves in accuracy for both this resolution and the 672^2 resolution, while degrading performance on lower resolutions (original 224^2 and lowest 160^2 resolutions), as shown in Figure 2 (dashed). Fine-tuned *sinpos* and CAPE on the 384^2 resolution outperform *abspos*, thanks to a better fine-tuning starting point. In that setting, *sinpos* and CAPE keep good generalization performance on nearby resolutions (224^2 and 672^2). While being seriously impacted on the 160^2 resolution, CAPE still outperforms others there.

No Positional Embedding With *nopos* model, we confirm [14]'s observation that positional embedding does not have a critical importance for ImageNet classification (see Figure 2). *Nopos* model's simplicity and generalization abilities make it a nice baseline for positional embedding study in computer vision. It has similar generalization accuracy than *abspos* on low 160^2 and high 672^2 resolutions, while CAPE outperforms *nopos* across the board. It is likely that *abspos* suffers from the embedding interpolation step on extreme resolutions. In contrast, both *sinpos* and CAPE have the advantage to naturally support different resolutions.

5.1.2 UniViT: Training Universal Transformer on Different Resolutions

As CAPE-based models can handle any image resolution, we train a single universal ViT model with CAPE, called UniViT, on randomly resized images (scaling is set to $\lambda = 1$). During training we resize all images in the same batch to a randomly chosen size, uniformly sampled in the range $[128, 320]^2$ with a step of 32^2. The rest of the training configuration remains the same as before. We compare UniViT against ViT models trained with either *abspos* or CAPE on each particular resolution, Figure 3. On a specific training resolution CAPE outperforms *abspos*, and generalizes better to other resolutions. UniViT performs well on all considered resolutions, always being close to the best performing single-resolution ViT model for any given resolution.

In Figure 4, we show UniViT throughput and accuracy with respect to input image resolution. Image resolution directly impacts throughput: computational complexity of attention is $O(N^4)$ for a $N \times N$ image. On 96^2 resolution UniViT handles similar throughput and accuracy as "tiny" ViT, while "small" UniViT has significantly higher accuracy (with identical throughput) on resolution 160^2.

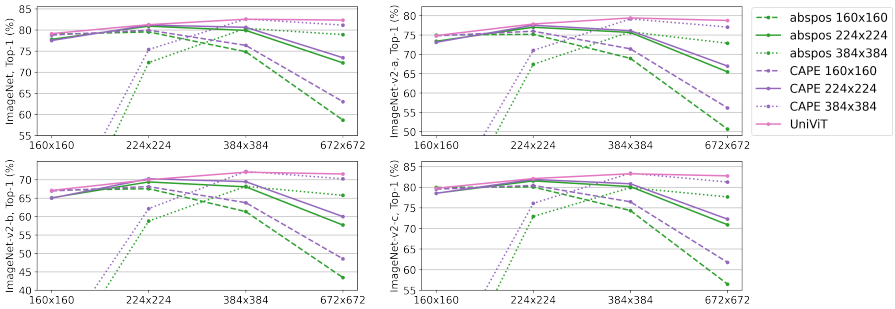

Figure 3: Top-1 accuracy on ImageNet and ImageNet-v2 for ViT models with either *abspos* or CAPE trained on each particular resolution and UniViT model trained on the mixture of resolutions.

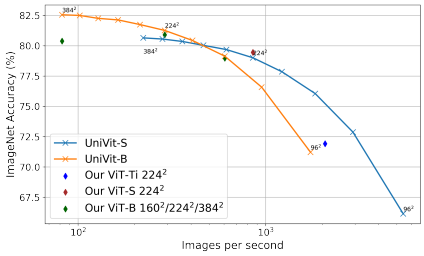

Figure 4: Dependence between throughput and accuracy on ImageNet at inference time under variation of image resolution. '-S' and '-Ti' refer to small and tiny architectures [14], respectively.

Table 1: Test time augmentation (TTA) results on ImageNet when predictions on resolutions r^2, $(r+32)^2$ and $(r-32)^2$ are combined. *Sinpos* and CAPE are trained on 224^2 resolution.

Model	r	Top-1 (%)		Top-5 (%)	
		-TTA	+TTA	-TTA	+ TTA
sinpos	224	81.39	81.53	95.51	95.64
CAPE	224	81.16	81.59	95.44	95.75
UniViT, sinpos	224	80.82	81.34	95.40	95.57
UniViT	224	81.26	81.64	95.56	95.71
UniViT, sinpos	384	82.31	82.44	96.04	96.14
UniViT	384	82.55	82.72	96.18	96.22

UniViT unlocks the possibility of dynamically adjusting throughput of a model in a production regime under heavy loads, a practical alternative to improving model throughput at inference time via decreasing model size.

We further improve UniViT accuracy performance by resizing each image to the corresponding optimal resolution for UniViT, as shown in Table 5 in Appendix B.2. We split ImageNet validation images into 8 bins, according to their size. By selecting an optimal resizing strategy in each bin we are able to improve top-1 accuracy to 82.92% (in comparison, UniVit has 81.26% on 224^2 and 82.55% on 384^2).

5.1.3 Resizing as Test Time Augmentation (TTA)

As both *sinpos* and CAPE-based models handle well different image resolutions, we propose to perform test time resolution augmentation when evaluating a single model. For TTA we average model's logits evaluated on three resolutions for the same image: r^2, $(r-32)^2$ and $(r+32)^2$, where r is either 224 or 384. As show in Table 1, ViT or UniViT models trained with either *sinpos* or CAPE embeddings get an accuracy boost with this test time augmentation.

5.2 Automatic Speech Recognition (ASR)

Recently it was shown that Transformer [39] architectures are state-of-the-art on different public benchmarks for ASR [45, 28, 26, 7].

Data We evaluate our models on several English speech datasets, both on in-domain data and out-of-domain data. We also analyze generalization of our models to long sequences. We consider two standard training benchmarks: Wall Street Journal (WSJ) [17, 25, 42], read speech with 81.5h of training data, and TED-LIUM v3 (TL) [21], oratory speech with 452h of training data. Besides these two datasets we use two other sets for evaluation only: LibriSpeech (LS) [30], read speech from audiobook recordings (we use only test sets with clean *test-clean* and noisy *test-other* speech);

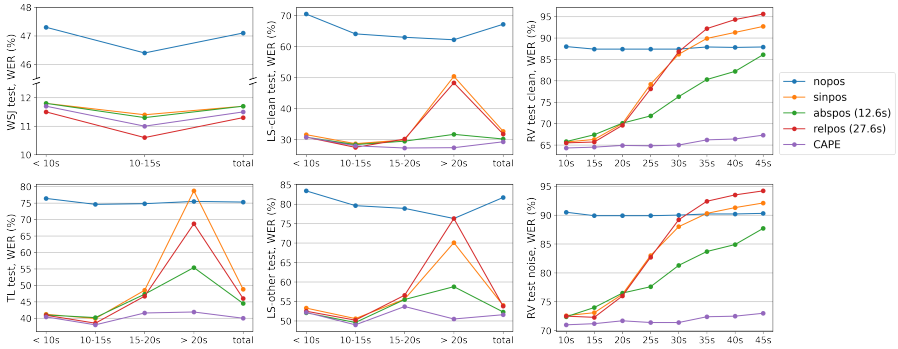

Figure 5: Word error rate for models trained on WSJ with different positional embeddings.

Robust Video (RV) is our in-house English video dataset, which is sampled from public social media videos and aggregated and de-identified before transcription. These videos contain a diverse range of speakers, accents, topics, and acoustic conditions making ASR difficult. The test sets are composed of *clean* and *noisy* subsets. Details on data and its statistics can be found in Appendix C.1.

Evaluation To evaluate the performance of our acoustic models on sequence lengths not seen at training time, we split all evaluation utterances by their duration T into the following bins: $T < 10$s, $T \in [10 - 15)$s, $T \in [15, 20)$s and $T >= 20$s. Our performance metric is word error rate (WER) (no language model was involved), reported for both each sequence size bin, and for the entire evaluation dataset. For RV data a hand-crafted segmentation is available, allowing us to evaluate on the same exact data, but segmented in different ways. More precisely, for RV data we prepare 8 sets where utterances have the following respective duration: $T = 10, 15, 20, 25, 30, 35, 40, 45$s.

Acoustic Model (AM) Training. All models are trained with Connectionist Temporal Classification [19]. SpecAugment [31] is used as data augmentation in training, and the network architecture follows [26]: the AM encoder is composed of a 1D convolution (kernel 7, stride 3) with a GLU activation, positional embedding and 36 4-heads Transformer blocks [39], finally followed by a linear layer which outputs a score for each target token. Our token set consists of 26 English alphabet letters, augmented with the apostrophe and a word boundary token (further details see in Appendix C.2).

Positional Embedding As for vision experiments, we evaluate *nopos*, *sinpos*, *abspos* and CAPE-based models. In addition, we evaluate models trained with relative positional embeddings: in that case, no absolute position is used, and learnable relative positional embeddings [36] (*relpos*) are trained in each Transformer block. We follow [26] to train an AM baseline with *relpos*. For models based on other positional embeddings, the training configuration remains identical. *Abspos* $\{\mathbf{E}(t)\}_{t=1}^{N}$ is set to cover 13.8s of context. At training/evaluation time for the longer sequences we define *abspos* as $\mathbf{E}(t) \equiv \mathbf{E}(t \mod N)$ for $t > N$. This extrapolation at training time leaves a chance to the acoustic model to generalize to the unseen (longer) sequence sizes. *Relpos* spans a large context, 26.8s to the left/right. CAPE's global shift covers 60s, while a local shift is set to its maximum to preserve the frames order; $\lambda_{max} = 1.1$ and $\lambda_{max} = 2$ for WSJ and TL, respectively.

5.2.1 Results

An acoustic model trained on WSJ with CAPE outperforms other positional embeddings on both public and RV data across different audio durations, as shown in Figure 5. A model trained on TL with CAPE outperforms *nopos* and *sinpos* on all data, outperforms *abspos* and *relpos* for audio longer than 20s, and behaves similarly on shorter durations (see Figure 6). On RV data, CAPE-based models perform uniformly well on different utterance sizes, including long utterances. In contrast, other embeddings-based models are seriously impacted when audio duration increases. Finally, CAPE does not have computational or parameters overheads compared to *relpos*.

No Positional Embedding As expected, *nopos* models (both WSJ and TL ones) perform similarly in WER across different audio duration. However, *nopos* TL model performs surprisingly well: it is

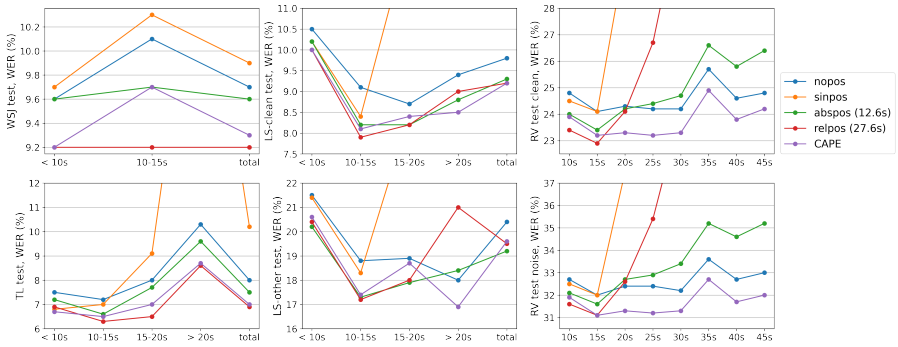

Figure 6: Word error rate for models trained on TED-LIUM v3 with different positional embeddings.

competitive to positional embeddings-based models on public data. On RV data, *nopos* TL model outperforms all other models, except CAPE when $T > 20$s. We perform ablations in Appendix C.4 to show that key ingredients are the CTC loss and large amount of data for this effect to occur.

CAPE as Augmentation CAPE can be viewed as a data augmentation performed on input data, which regularizes the trained model. We demonstrate this by training on TL with either *sinpos* or CAPE and with/without SpecAugment (no other augmentations are used), see Figure 7. Baseline *sinpos* without any augmentation performs the worst with a large gap. Including either CAPE or SpecAugment decreases the WER significantly. SpecAugment is more beneficial due to its domain relevance. Combining together CAPE and SpecAugment further decreases the WER and ends up with the best model, showing that augmentations are complementary to each other.

5.2.2 Padding-free ASR with CAPE and Variable STFT Hop Distance

In ASR, when batching utterances of different sizes, one often relies on padding tokens. We propose here instead to perform time stretching augmentation on all utterances in the batch, such that they have the same length. We perform this augmentation by tuning the short-time Fourier Transform (STFT) hop distance when computing the audio features. CAPE embeddings remain tied to the original timestamp of the audio. Models trained either on WSJ or TL show better WER across the board, compared to models trained with a padding approach, as shown in Appendix C.3. CAPE-based models outperform *sinpos* models. We found this padding-free approach convenient, as it *alleviates ther implementation of special cases to handle padding tokens* in ASR models.

5.3 Machine Translation (MT)

Our MT experiments follow the recent results with Transformers combined with a new initialization scheme (ADMIN) [27]. This approach allows training very deep Transformers for MT and reaches state-of-the-art performance. We did not implement back-translation or other domain-specific augmentations.

Data and Models Training Experiments are conducted on standard WMT'14 English-French (FR) and English-German (DE) benchmarks. For both benchmarks we follow [27]: for FR we use a 40k subword vocabulary, and evaluate on the provided 'valid' file for validation and newstest14 for test. On DE, we consider a 32K subword vocabulary, newstest2013 for validation, and newstest2014 for test. We reproduce results from [27] by training a *sinpos*-based model with 6L-6L, 18L-18L for DE and 6L-6L for FR encoder-decoder layers with ADMIN. Training configuration stays the same for other positional embedding-based models, other than the positional embeddings being either *abspos* or CAPE in both encoder and decoder layers. For CAPE to have some correspondence in positions of source and target sentences, we first scale positions of source language by a factor $\alpha = \frac{\text{\# tokens in target corpus}}{\text{\# tokens in source corpus}} \in \mathbb{R}$ so that positions interval is loosely matched between source and target sentences. For each training sample we then apply the same global shift and scaling for source and target positions. Local shifts for source/target positions are independently sampled from $\mathcal{U}(-0.5, 0.5)$.

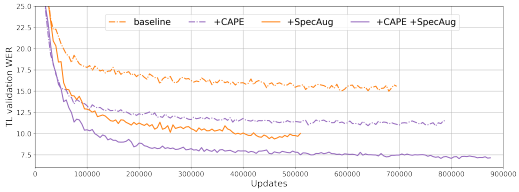

Figure 7: Validation WER for models trained with different augmentations: 'baseline' is a model with *sinpos*, '+CAPE' adds CAPE's global and local shifts, '+SpecAug' adds SpecAugment.

Table 2: BLEU on WMT'14 benchmarks.

Model	Lang.	Valid BLEU	Test BLEU
sinpos, 6L-6L	DE	26.92	27.54
abspos, 6L-6L	DE	26.69	27.33
CAPE, 6L-6L	DE	27.03	27.81
sinpos, 18L-18L	DE	27.12	27.9
abspos, 18L-18L	DE	27.25	28.24
CAPE, 18L-18L	DE	27.29	28.37
sinpos, 6L-6L	FR	47.25	41.03
abspos, 6L-6L	FR	47.18	41.17
CAPE, 6L-6L	FR	47.22	41.59

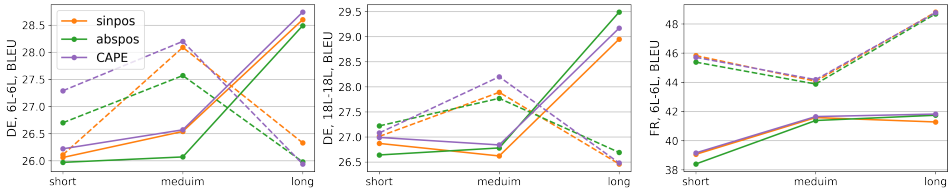

Figure 8: BLEU on WMT'14 benchmarks for short, medium and long sentences on validation (dashed) and test (solid) sets: 6L-6L DE (left), 18L-18L DE (middle) and 6L-6L FR (right).

Evaluation We select the best checkpoint according to BLEU on the validation set, using a beam size 4 for DE and 5 for FR. We also split test sets by sentence length for target language using 33% and 66% percentiles to have 'short', 'medium' and 'long' sentences, and report BLEU scores on each subset independently. Following convention, BLEU is computed by `multi-bleu.perl` via the standardized tokenization of the publicly-accessible dataset.

Results We show in Table 2 a comparison between positional embeddings for different encoder-decoder architectures on WMT'14 benchmarks. CAPE embeddings outperform *sinpos* and *abspos* on all settings. In Figure 8 we also report BLEU comparison on validation and test sets for short, medium and long sentences. From our experiments we observe poor correlation between validation and test sets for DE. On validation sets CAPE mostly behaves similar and sometimes outperforms others. On test sets CAPE is the best on short and middle sentences, while on long sentences it outperforms others for both DE and FR 6L-6L, and outperforms *sinpos* for DE 18L-18L.

6 Discussion and Conclusion

Encoding positional information is a key component of attention-based models. Poor generalization of absolute sinusoidal positional embeddings led to many work investigating ways to encode relevant positional information. Existing solutions are often modality-specific, non-trivial to implement, while incurring computational overheads.

We demonstrated that existing positional embeddings may generalize poorly in certain conditions across different domains. We introduced a simple and efficient continuous augmented positional embedding, CAPE, which preserves some information about relative token positions. Thanks to its continuous nature, CAPE allows augmentations which were previously not possible. CAPE also makes models more flexible both at training and inference. It generalizes well to input sizes not seen during training across a variety of domains. We expect emergence of new training and production pipelines that leverage the adjustable throughput property when tuning the input size. Going further, CAPE-based architectures are free from baked-in restrictions on patches positions: these could overlap, or be sparse for example. Finally, introduced ideas can be combined with relative positional embedding like [10] and [37] to limit over-fitting to exact relative positions.

Limitations Our CAPE embeddings applies only to attention-based models, and no testing was performed outside described modalities.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[7] William Chan, Daniel Park, Chris Lee, Yu Zhang, Quoc Le, and Mohammad Norouzi. Speechstew: Simply mix all available speech recognition data to train one large neural network. *arXiv preprint arXiv:2104.02133*, 2021.

[8] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[9] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. *Advances in Neural Information Processing Systems*, 33, 2020.

[10] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[13] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE, 2018.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12, 2011.

[16] A. Fan, E. Grave, and A. Joulin. Reducing transformer depth on demand with structured dropout. In *ICML*, 2020.

[17] J. Garofolo, D. Graff, D. Paul, and D. Pallett. CSR-I (WSJ0) complete LDC93S6A. *Web Download. Philadelphia: LDC*, 1993.

[18] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021.

[19] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

[20] Anmol Gulati et al. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech*, 2020.

[21] F. Hernandez et al. TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *SPECOM*, 2018.

[22] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8126–8135. IEEE Computer Society, 2020.

[23] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *International Conference on Learning Representations*, 2018.

[24] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking the positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*, 2020.

[25] LDC and NIST Multimodal Information Group. CSR-II (WSJ1) complete LDC94S13A. *Web Download. Philadelphia: LDC*, 1994.

[26] Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Paden Tomasello, Jacob Kahn, Gilad Avidov, Ronan Collobert, and Gabriel Synnaeve. Rethinking evaluation in asr: Are our models robust enough? *arXiv preprint arXiv:2010.11745*, 2020.

[27] Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*, 2020.

[28] Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer. Transformers with convolutional context for asr. *arXiv preprint arXiv:1904.11660*, 2019.

[29] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

[30] V. Panayotov et al. Librispeech: an ASR corpus based on public domain audio books. In *ICASSP*, 2015.

[31] D. S. Park et al. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, 2019.

[32] D. Povey and other. The Kaldi speech recognition toolkit. In *ASRU*, 2011.

[33] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[34] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[36] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, 2018.

[37] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

[38] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[39] A. Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[40] Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. On position embeddings in bert. In *International Conference on Learning Representations*, volume 2, pages 12–13, 2021.

[41] Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al. Transformer-based acoustic modeling for hybrid speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6874–6878. IEEE, 2020.

[42] P. C. Woodland et al. Large vocabulary continuous speech recognition using HTK. In *ICASSP*, 1994.

[43] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

[44] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6022–6031. IEEE, 2019.

[45] Albert Zeyer, Parnia Bahar, Kazuki Irie, Ralf Schlüter, and Hermann Ney. A comparison of transformer and lstm encoder decoder models for asr. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 8–15. IEEE, 2019.

[46] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[47] Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu. Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2010.10504*, 2020.

[48] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34-07, pages 13001–13008, 2020.

[49] Pan Zhou, Ruchao Fan, Wei Chen, and Jia Jia. Improving generalization of transformer for speech recognition with parallel schedule sampling and relative positional embedding. *arXiv preprint arXiv:1911.00203*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Details in Section 6.

(c) Did you discuss any potential negative societal impacts of your work? [No]

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [N/A]

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include python implementation of CAPE in the Appendix A. All our experiments are based on top of the open-sourced code where used state-of-the-art recipes for machine translation, vision Transformer and speech recognition were published. We provide these URLs in the Appendix B.1, C.2, and D.1. The changes we made into these public codebases are included in supplemental material and will be made publicly available upon acceptance.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] . This information is provided in the main text and also in the Appendix B.1, C.1, C.2, C.3, D.1

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We look at results across different domains and models types, which presents variability in a different format than error bars.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We specify all compute details in Appendix B.1, C.2 and D.1.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We provide bibliographic references or URLs for the relevant resources.

(b) Did you mention the license of the assets? [No] Provided bibliographic references and URLs simply allow to find out about the licenses of the various asset used in the paper.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide our code in the supplemental material, and we will also release it upon publication.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We are using existing assets in compliance with their respective licenses, that do not require direct consent by the creators. We also use our in-house English video dataset in according to our data policy: it is aggregated and deidentified before transcription.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Our work does not contain or use personally identifiable information or offensive content. Specifically, we discuss this issue only for our in-house English video dataset in Section 5.2

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]